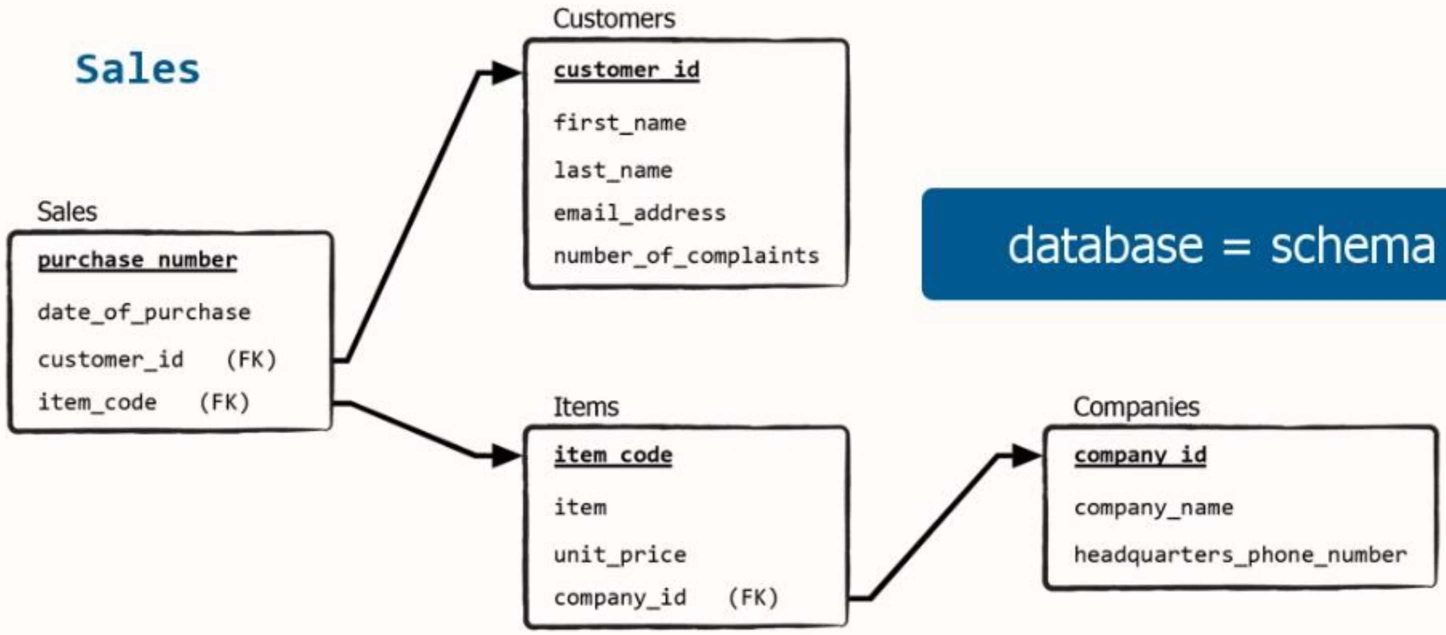
Creating a Database - Part I

Sales



database = schema

DDL for creating database

- CREATE DATABASE [if not exists] database_name;

[indicates optional statements]

SQL is case insensitive

```
CREATE DATABASE IF NOT EXISTS Sales;
```

or

```
CREATE SCHEMA IF NOT EXISTS Sales;
```

Running: CTRL + SHIFT + ENTER

Data types

- If not numeric, then ‘’ must be used.
- Also there are BLOB data type: Binary Large OBject: 0011…

docx. xlsx, jpeg etc. For example, for photos.

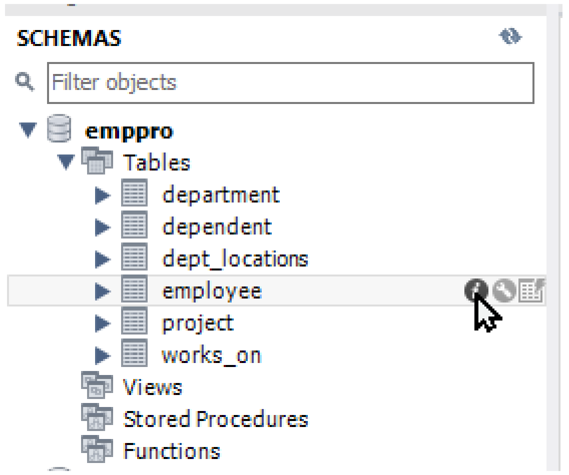
Postgre, Microsoft sql vb all implement same set of variable types.

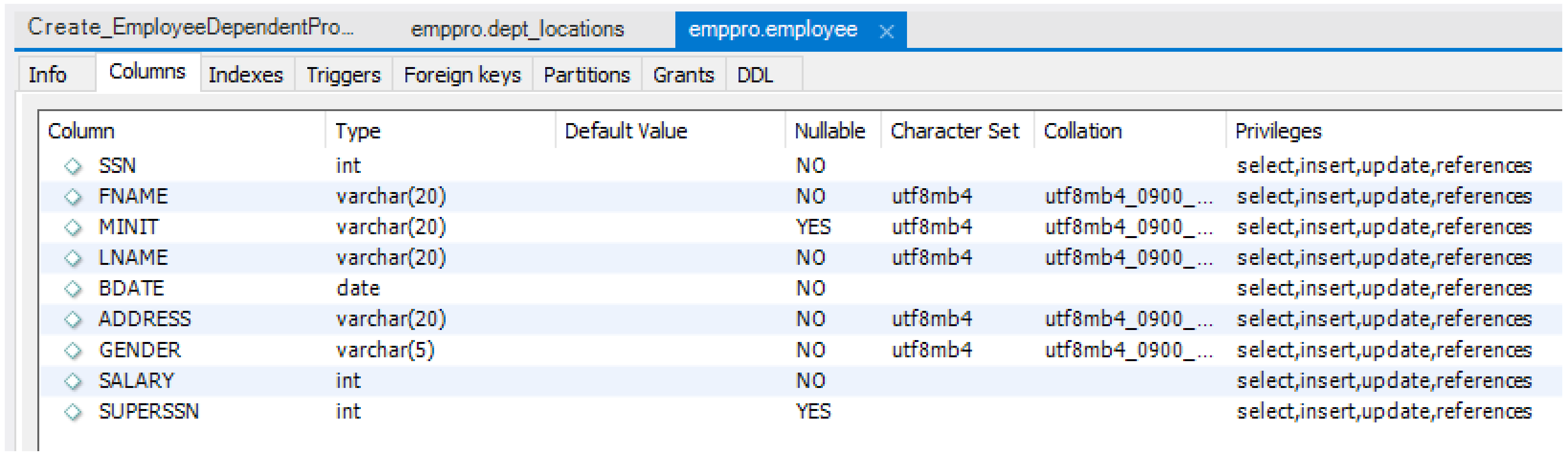
Specifying the database to which we refer

- Syntax:
- Set a default database:
  - USE dbasename;
- Specify database in SQL query using «dot» operator:
  - SELECT * FROM dbasename.dbasetable;

Defining a table

```
CREATE table EMPLOYEE
(
SSN INT NOT NULL PRIMARY KEY,
FNAME VARCHAR(20) NOT NULL,
MINIT VARCHAR(20),
LNAME VARCHAR(20) NOT NULL,
BDATE DATE NOT NULL,
ADDRESS VARCHAR(20) NOT NULL,
GENDER VARCHAR(5) NOT NULL,
SALARY INT NOT NULL,
SUPERSSN INT
);
```





Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
SSN	int		NO			select,insert,update,references
FNAME	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
MINIT	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
LNAME	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
BDATE	date		NO			select,insert,update,references
ADDRESS	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
GENDER	varchar(5)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
SALARY	int		NO			select,insert,update,references
SUPERSSN	int		YES			select,insert,update,references

Adding constraints

- Specific rules, or limits that we define in tables
  - NOT NULL
  - PRIMARY KEY
  - FOREIGN KEY
  - UNIQUE KEY
- CASCADE DELETE: If a value from the parent table's primary key is removed, all corresponding records from the child table will be removed as well.
  - Direction is important

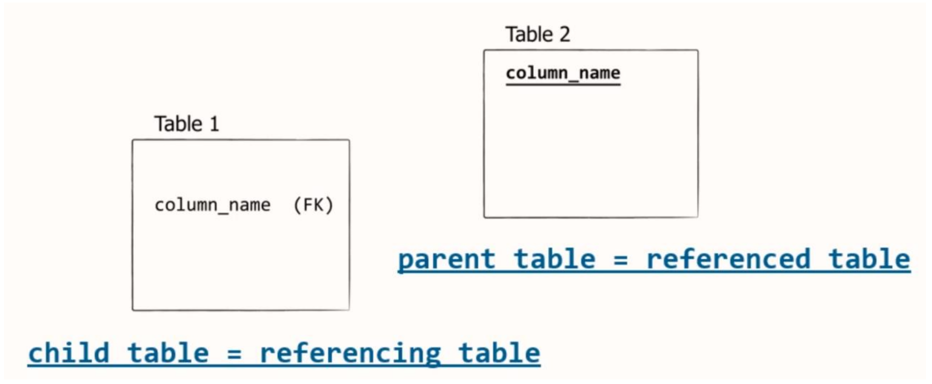
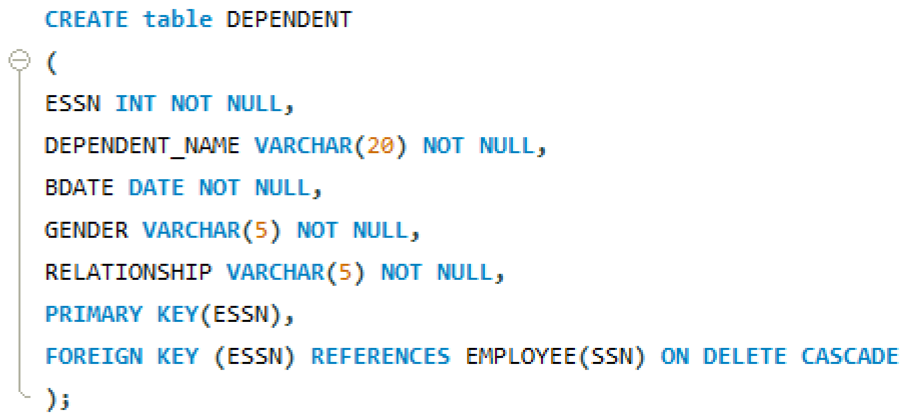


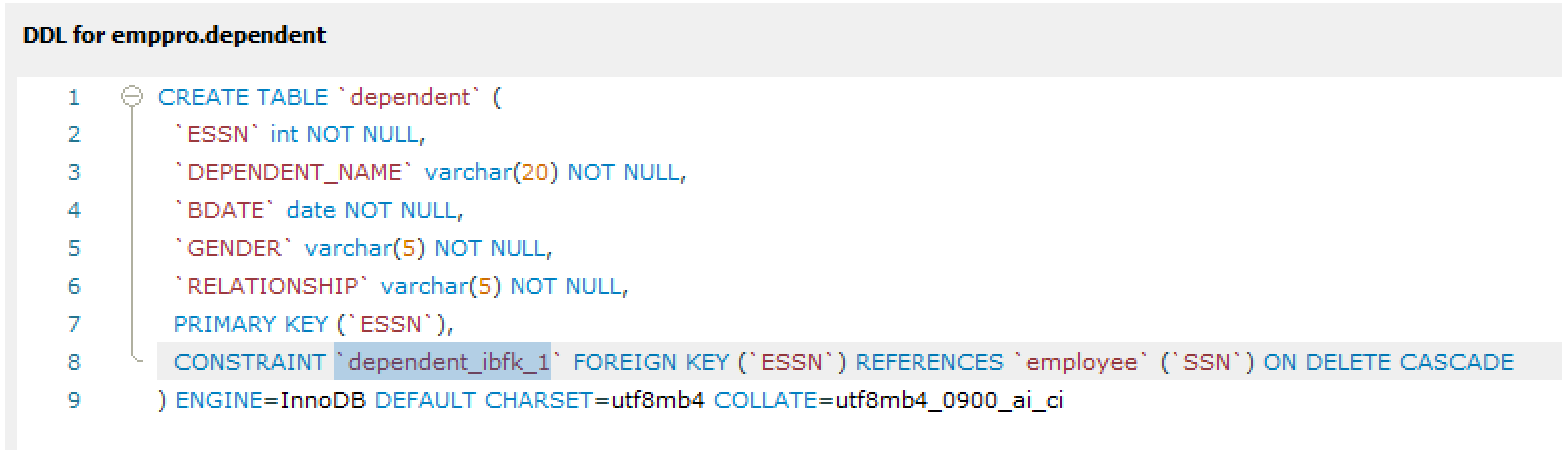
Table 2
column_name
Table 1
column_name (FK)
parent table = referenced table
child table = referencing table



```
CREATE table DEPENDENT
(
ESSN INT NOT NULL,
DEPENDENT_NAME VARCHAR(20) NOT NULL,
BDATE DATE NOT NULL,
GENDER VARCHAR(5) NOT NULL,
RELATIONSHIP VARCHAR(5) NOT NULL,
PRIMARY KEY(ESSN),
FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE
);
```

Adding/removing the FK constraint later on

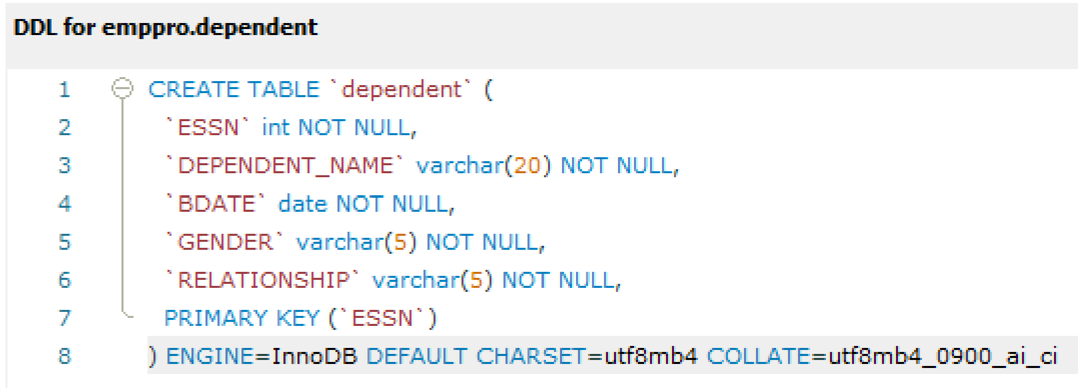
```
ALTER TABLE DEPENDENT
ADD FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE;
```



DDL for emppro.dependent

```
CREATE TABLE `dependent` (
  `ESSN` int NOT NULL,
  `DEPENDENT_NAME` varchar(20) NOT NULL,
  `BDATE` date NOT NULL,
  `GENDER` varchar(5) NOT NULL,
  `RELATIONSHIP` varchar(5) NOT NULL,
  PRIMARY KEY (`ESSN`),
  CONSTRAINT `dependent_ibfk_1` FOREIGN KEY (`ESSN`) REFERENCES `employee` (`SSN`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

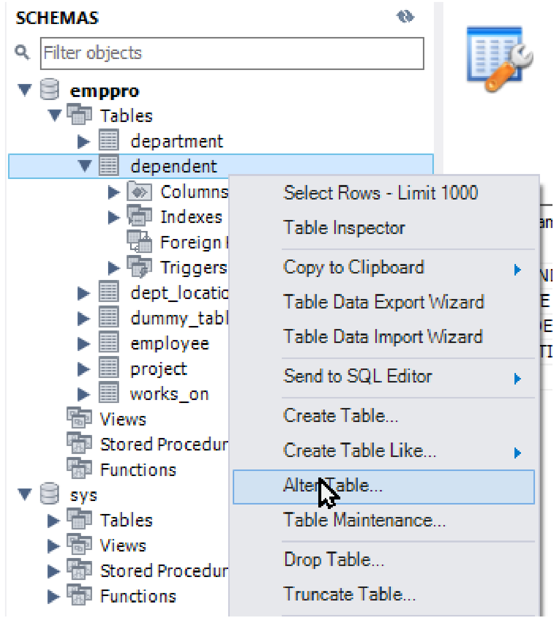
```
ALTER TABLE DEPENDENT
ADD FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE;
```



DDL for emppro.dependent

```
CREATE TABLE `dependent` (
  `ESSN` int NOT NULL,
  `DEPENDENT_NAME` varchar(20) NOT NULL,
  `BDATE` date NOT NULL,
  `GENDER` varchar(5) NOT NULL,
  `RELATIONSHIP` varchar(5) NOT NULL,
  PRIMARY KEY (`ESSN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Adding/removing the FK constraint later on-II



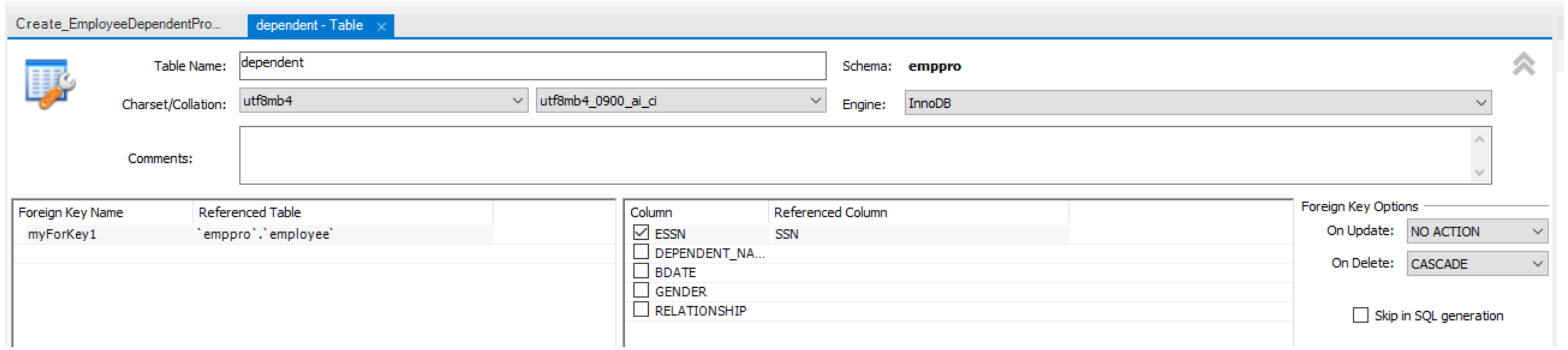
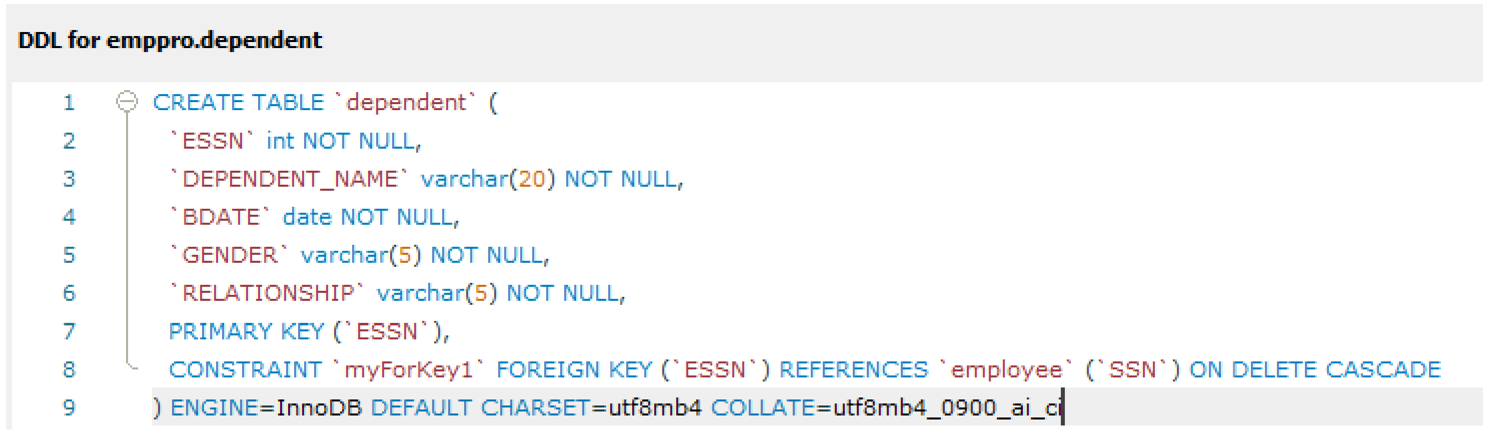


Table Name: dependent
Schema: emppro
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci
Engine: InnoDB
Comments:

Foreign Key Name	Referenced Table
myForKey1	\`emppro\`.\`employee\`

Column	Referenced Column
✓ ESSN	SSN
☐ DEPENDENT_NA...	
☐ BDATE	
☐ GENDER	
☐ RELATIONSHIP	

Foreign Key Options
On Update: NO ACTION
On Delete: CASCADE
☐ Skip in SQL generation

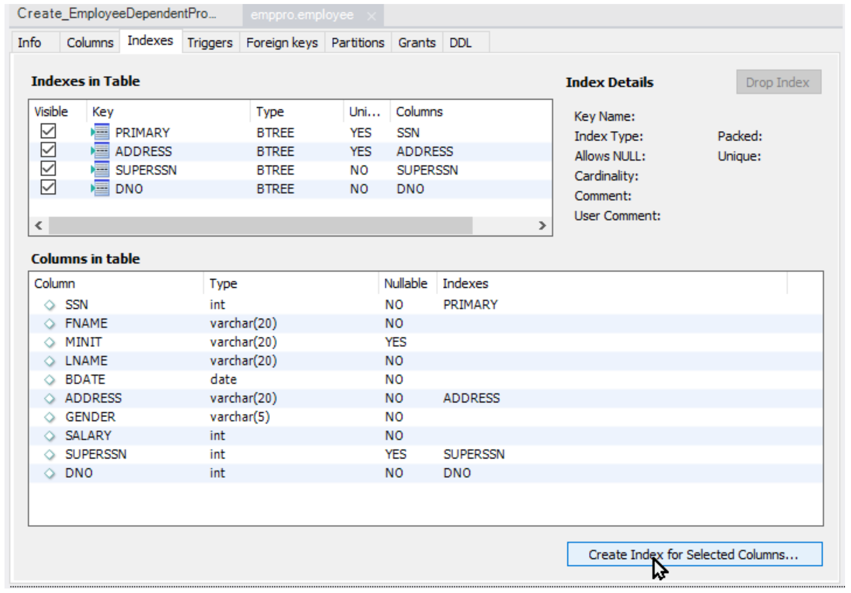


DDL for emppro.dependent

```sql
CREATE TABLE `dependent` (
  `ESSN` int NOT NULL,
  `DEPENDENT_NAME` varchar(20) NOT NULL,
  `BDATE` date NOT NULL,
  `GENDER` varchar(5) NOT NULL,
  `RELATIONSHIP` varchar(5) NOT NULL,
  PRIMARY KEY (`ESSN`),
  CONSTRAINT `myForKey1` FOREIGN KEY (`ESSN`) REFERENCES `employee` (`SSN`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Adding constraints

- Specific rules, or limits that we define in tables
  - NOT NULL
  - PRIMARY KEY
  - FOREIGN KEY
  - UNIQUE KEY
- Cannot insert duplicate values for this column.
- But, can still be NULL



Indexes in Table

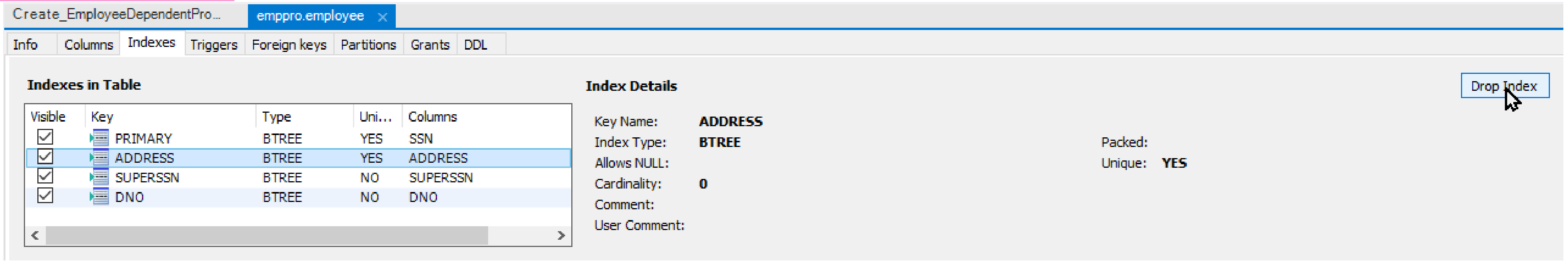
Visible	Key	Type	Uni...	Columns
☑	PRIMARY	BTREE	YES	SSN
☑	ADDRESS	BTREE	YES	ADDRESS
☑	SUPERSSN	BTREE	NO	SUPERSSN
☑	DNO	BTREE	NO	DNO

Index Details

Key Name:
Index Type:
Allows NULL:
Cardinality:
Comment:
User Comment:
Packed:
Unique:

Columns in table

Column	Type	Nullable	Indexes
SSN	int	NO	PRIMARY
FNAME	varchar(20)	NO	
MINIT	varchar(20)	YES	
LNAME	varchar(20)	NO	
BDATE	date	NO	
ADDRESS	varchar(20)	NO	ADDRESS
GENDER	varchar(5)	NO	
SALARY	int	NO	
SUPERSSN	int	YES	SUPERSSN
DNO	int	NO	DNO



Indexes in Table

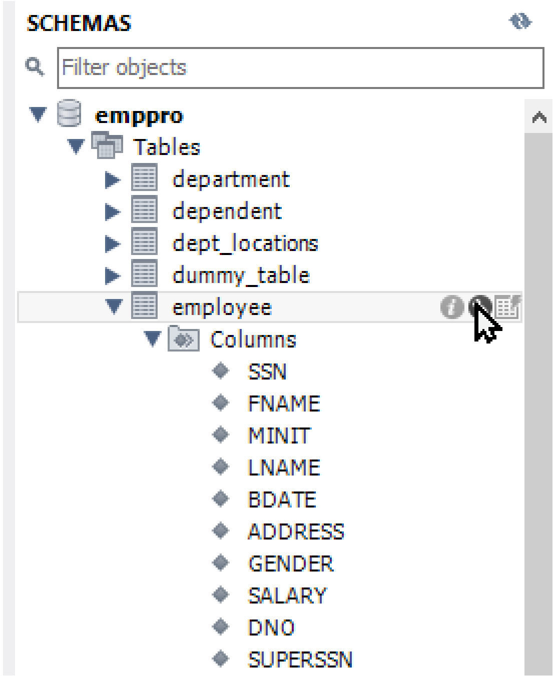
Visible	Key	Type	Uni...	Columns
☑	PRIMARY	BTREE	YES	SSN
☑	ADDRESS	BTREE	YES	ADDRESS
☑	SUPERSSN	BTREE	NO	SUPERSSN
☑	DNO	BTREE	NO	DNO

Index Details

Key Name: **ADDRESS**
Index Type: **BTREE**
Allows NULL:
Cardinality: **0**
Comment:
User Comment:
Packed:
Unique: **YES**

Adding constraints: default values

```
ALTER TABLE EMPLOYEE
ALTER COLUMN SALARY SET DEFAULT 10000;
```



SCHEMAS

Filter objects

- emppro
  - Tables
    - department
    - dependent
    - dept_locations
    - dummy_table
    - employee
      - Columns
        - SSN
        - FNAME
        - MINIT
        - LNAME
        - BDATE
        - ADDRESS
        - GENDER
        - SALARY
        - DNO
        - SUPERSSN

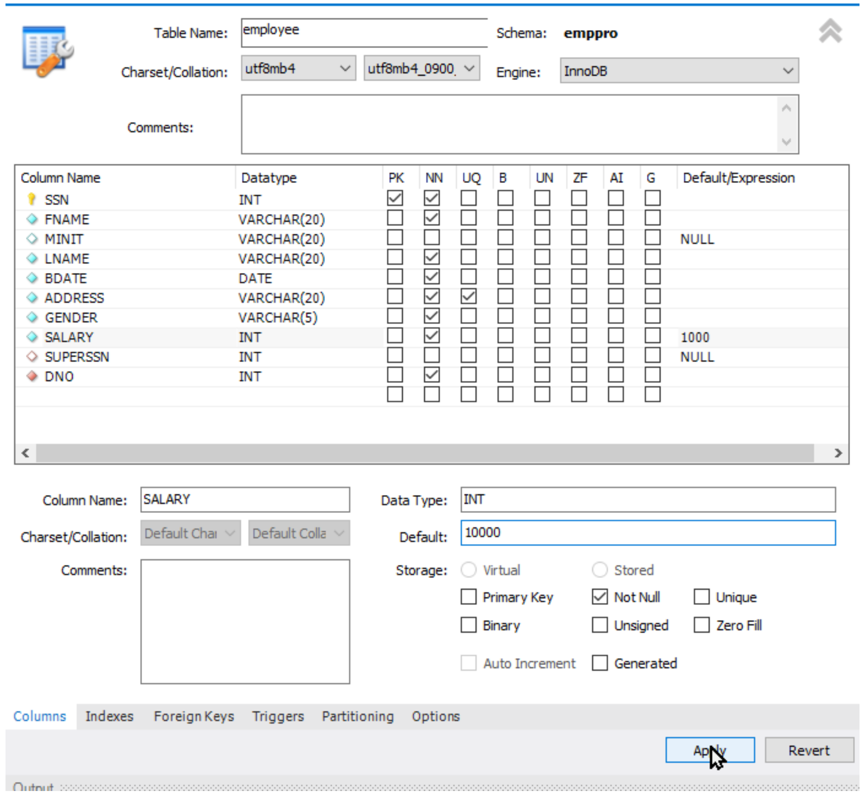


Table Name: employee Schema: emppro

Charset/Collation: utf8mb4 utf8mb4_0900_ Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
SSN	INT	✓	✓							
FNAME	VARCHAR(20)		✓							
MINIT	VARCHAR(20)									NULL
LNAME	VARCHAR(20)		✓							
BDATE	DATE		✓							
ADDRESS	VARCHAR(20)		✓	✓						
GENDER	VARCHAR(5)		✓							
SALARY	INT		✓							1000
SUPERSSN	INT									NULL
DNO	INT		✓							

Column Name: SALARY Data Type: INT

Charset/Collation: Default Char Default Colla Default: 10000

Comments: Storage: Virtual Stored

Primary Key [✓ Not Null] Unique

Binary Unsigned Zero Fill

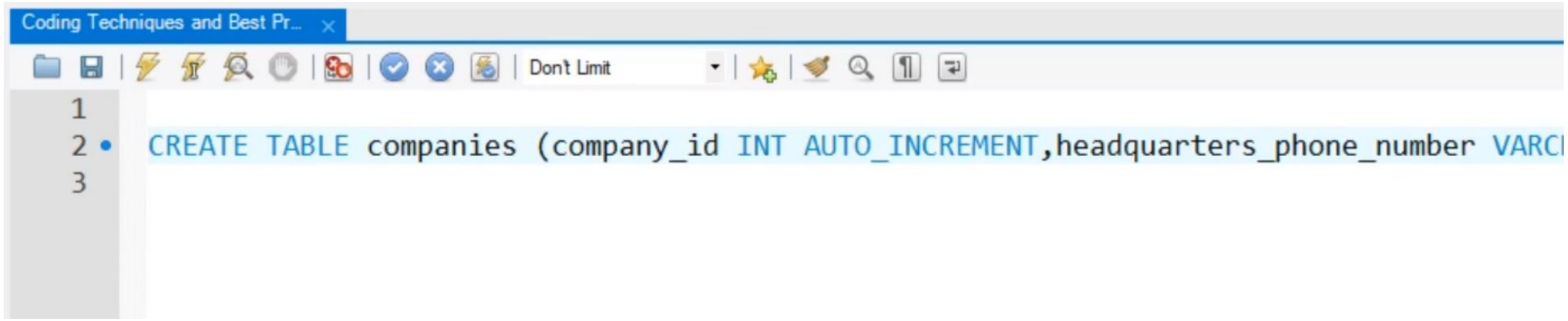
Auto Increment Generated

```
ALTER TABLE EMPLOYEE
ALTER COLUMN SALARY DROP DEFAULT;
```

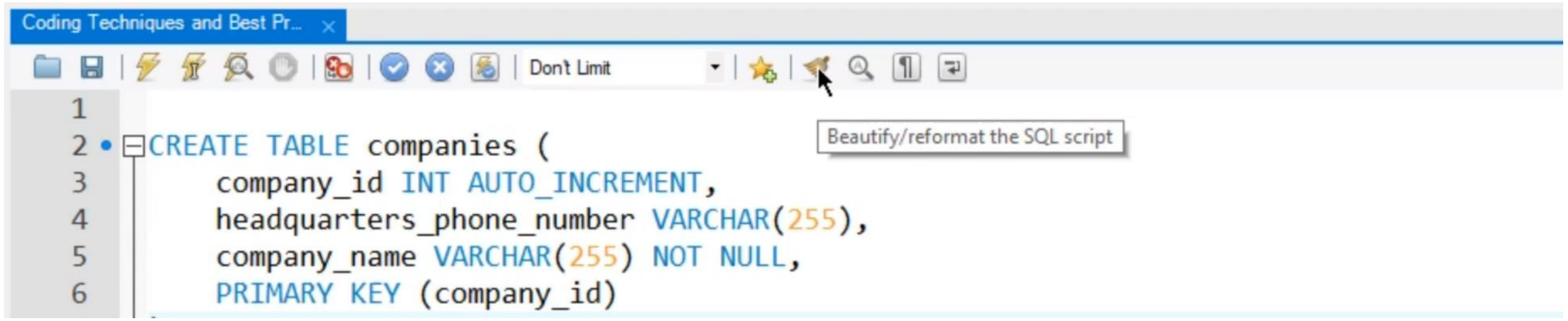
Conventions

SQL

- Naming convention:
  - productName
  - product_name
  - strProductName
  - No «space» between words
  - Don’t use «Product_unique_name» (too long, too limiting)
- Indentation
  - Use CTRL+B



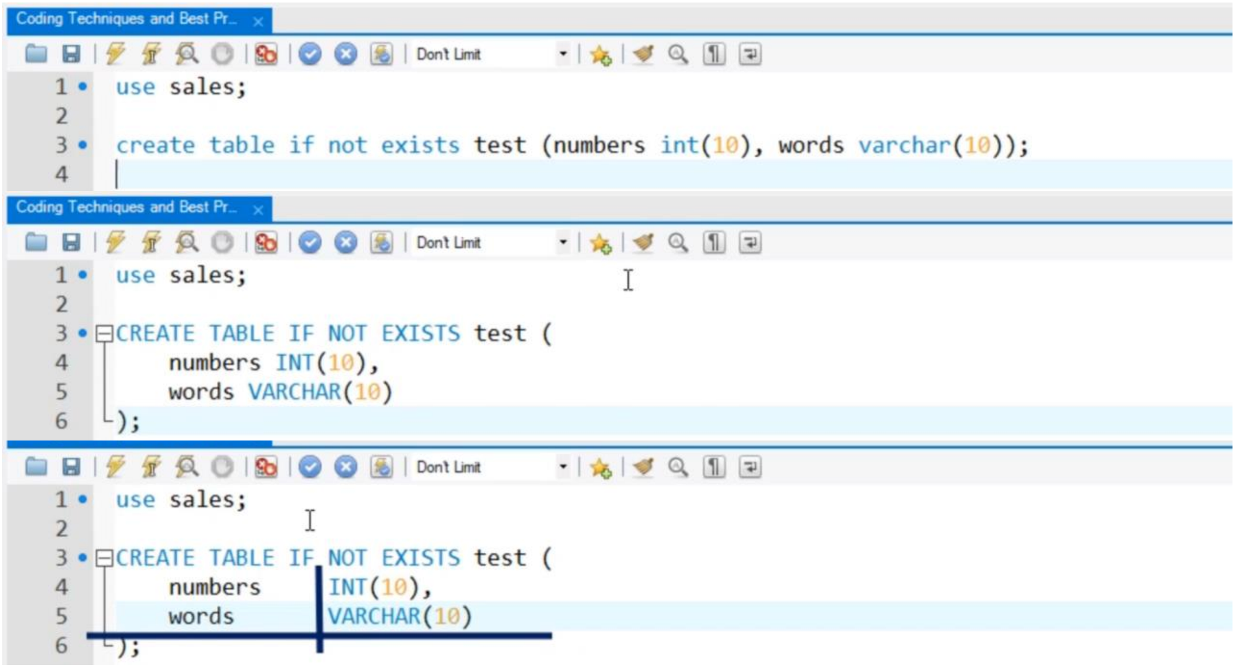
```

CREATE TABLE companies (company_id INT AUTO_INCREMENT,headquarters_phone_number VARC
```


Coding Techniques and Best Pr...
Don't Limit
Beautify/reformat the SQL script

```

CREATE TABLE companies (
    company_id INT AUTO_INCREMENT,
    headquarters_phone_number VARCHAR(255),
    company_name VARCHAR(255) NOT NULL,
    PRIMARY KEY (company_id)
```

Commenting and indentation

- Indentation:



Coding Techniques and Best Pr...

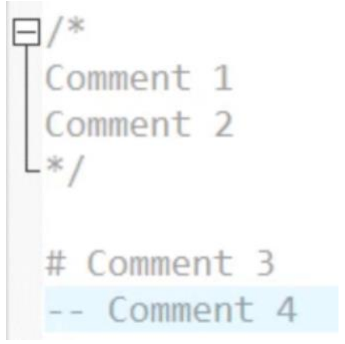
```
use sales;

create table if not exists test (numbers int(10), words varchar(10));

```

Coding Techniques and Best Pr...

```
use sales;

CREATE TABLE IF NOT EXISTS test (
    numbers INT(10),
    words VARCHAR(10)
);
```

```
use sales;

CREATE TABLE IF NOT EXISTS test (
    numbers     INT(10),
    words       VARCHAR(10)
);
```

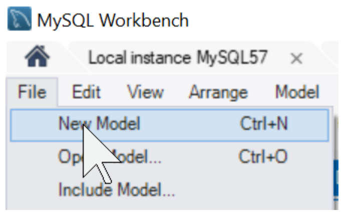
- Comments:

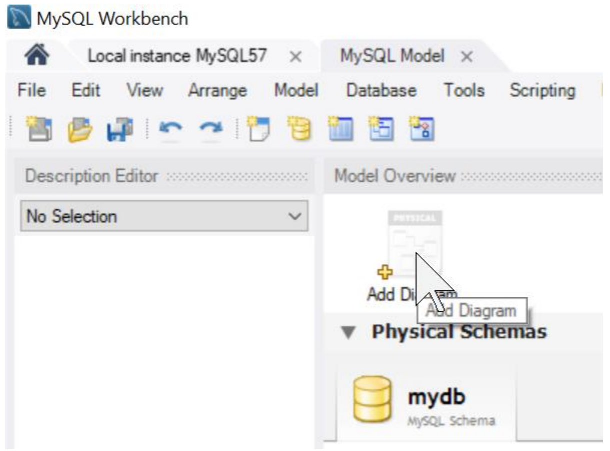


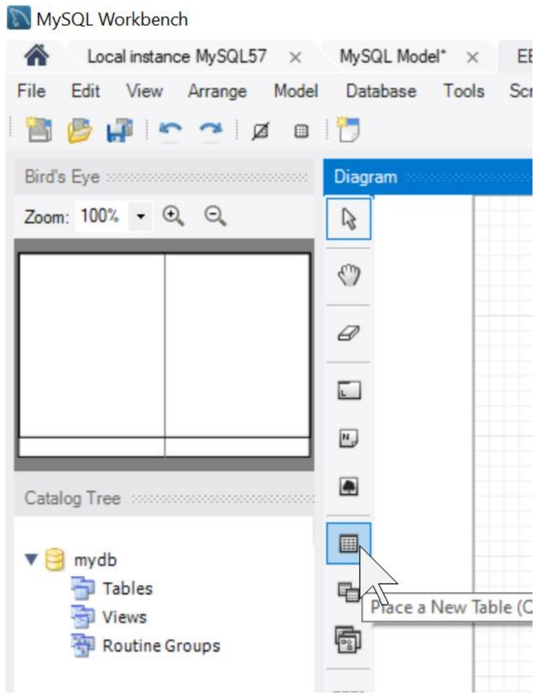
```
/*
Comment 1
Comment 2
*/

# Comment 3
-- Comment 4
```

Creating database and entering data-1

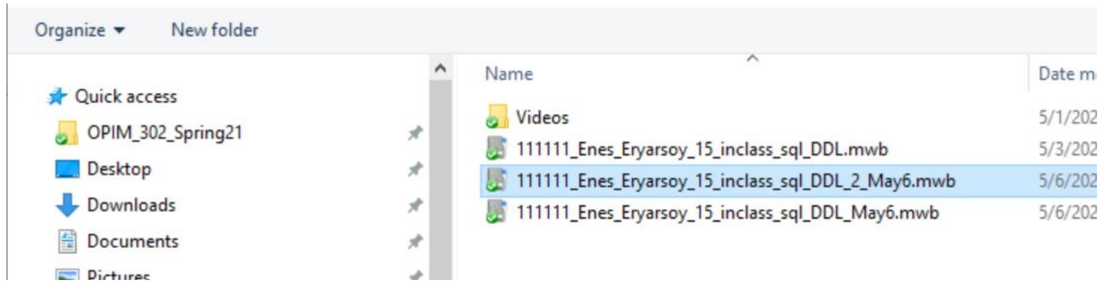
- Step 1: create your model:





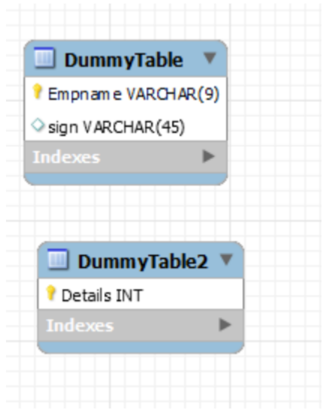


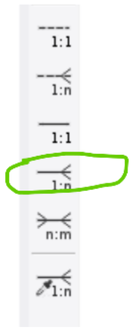
- Or load an existing one:

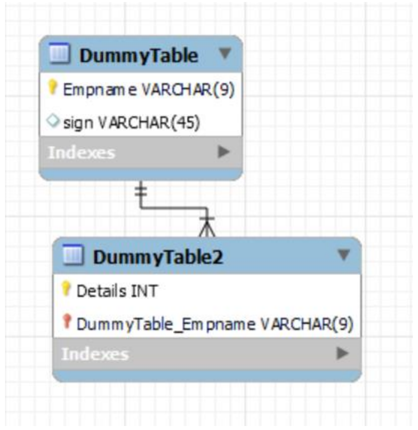


Name	Date m
Videos	5/1/202
111111_Enes_Eryarsoy_15_inclass_sql_DDL.mwb	5/3/202
111111_Enes_Eryarsoy_15_inclass_sql_DDL_2_May6.mwb	5/6/202
111111_Enes_Eryarsoy_15_inclass_sql_DDL_May6.mwb	5/6/202

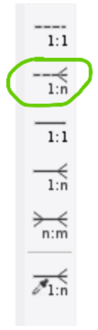
Connecting tables

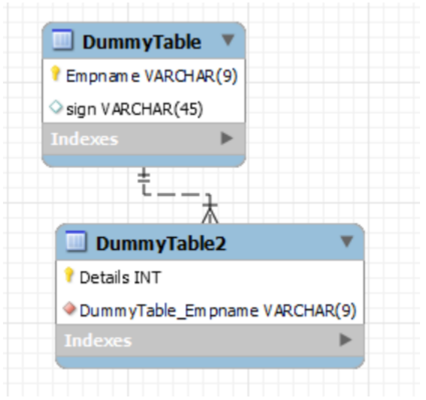






Identifying:
•The foreign key is also part of the primary key of the child table.
•This typically means that the child table cannot be uniquely identified without the parent table.

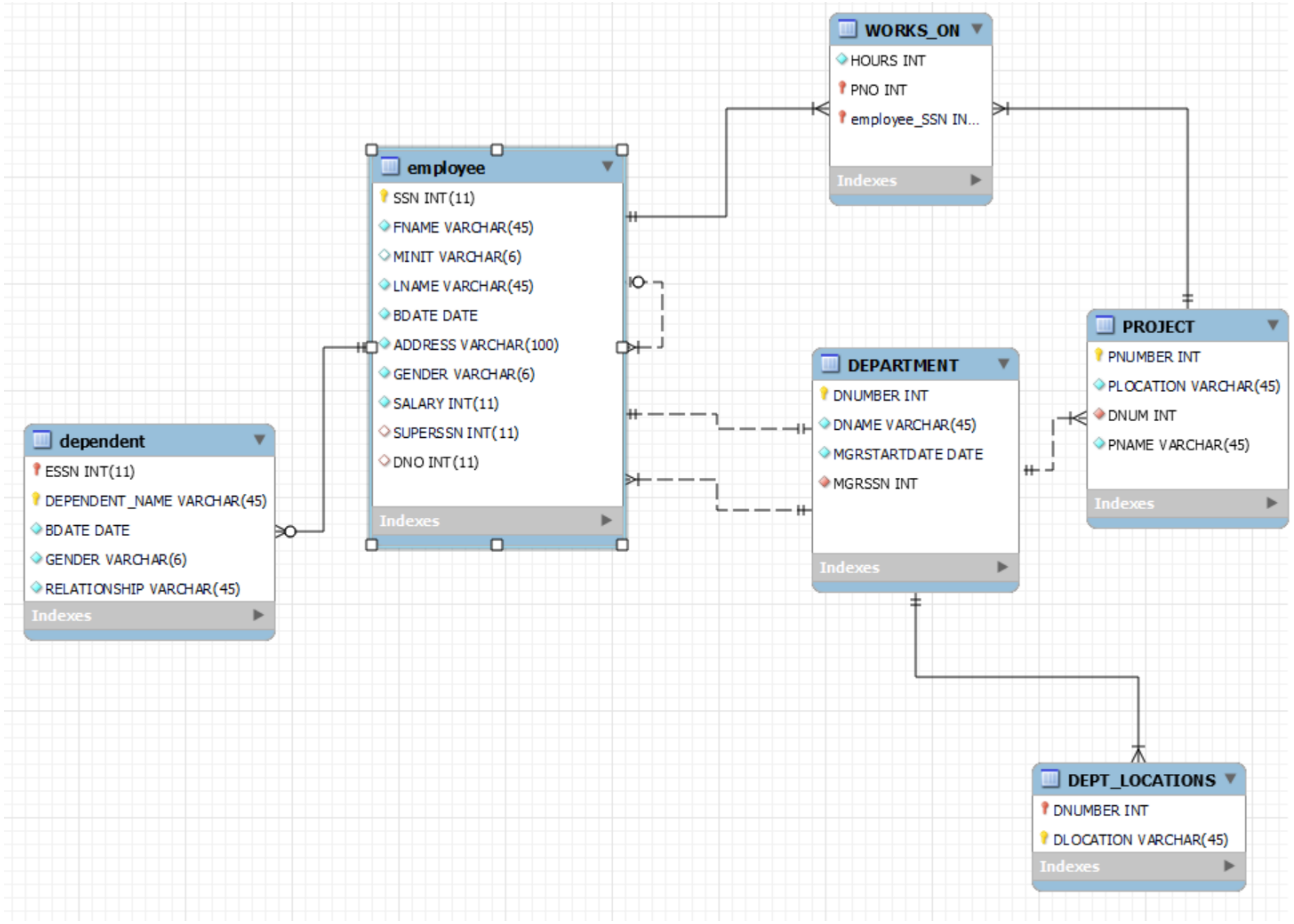




Non identifying:

When connecting: first click on the many side.

Check your model



🔑🔑 Key: (Part of) Primary Key

◆◆ Filled Diamond: NOT NULL

◇◇ Not filled Diamond: NULL

🔑◆◇ Red colored: (Part of) Foreign key

◆◇ Blue lined Diamond: Simple attribute (no key)

Can be combined for example:

🔑 is a Red colored Key so it's a Primary Key which is also a Foreign Key

🔑 is a Yellow (non Red) Key so it's only a Primary Key

◆ is a blue lined filled diamond so it's a NOT NULL simple attribute

◆ is a red colored filled diamond so it's a NOT NULL Foreign Key

◇ is a blue lined not filled diamond so it's a simple attribute which can be NU

◇ is a red colored not filled diamond so it's a Foreign Key which can be NUL

Identifying vs non-identifying

- Identifying relationships exist when the primary key of the parent entity is included in the primary key of the child entity. On the other hand, a non-identifying relationship exists when the primary key of the parent entity is included in the child entity but not as part of the child entity's primary key.

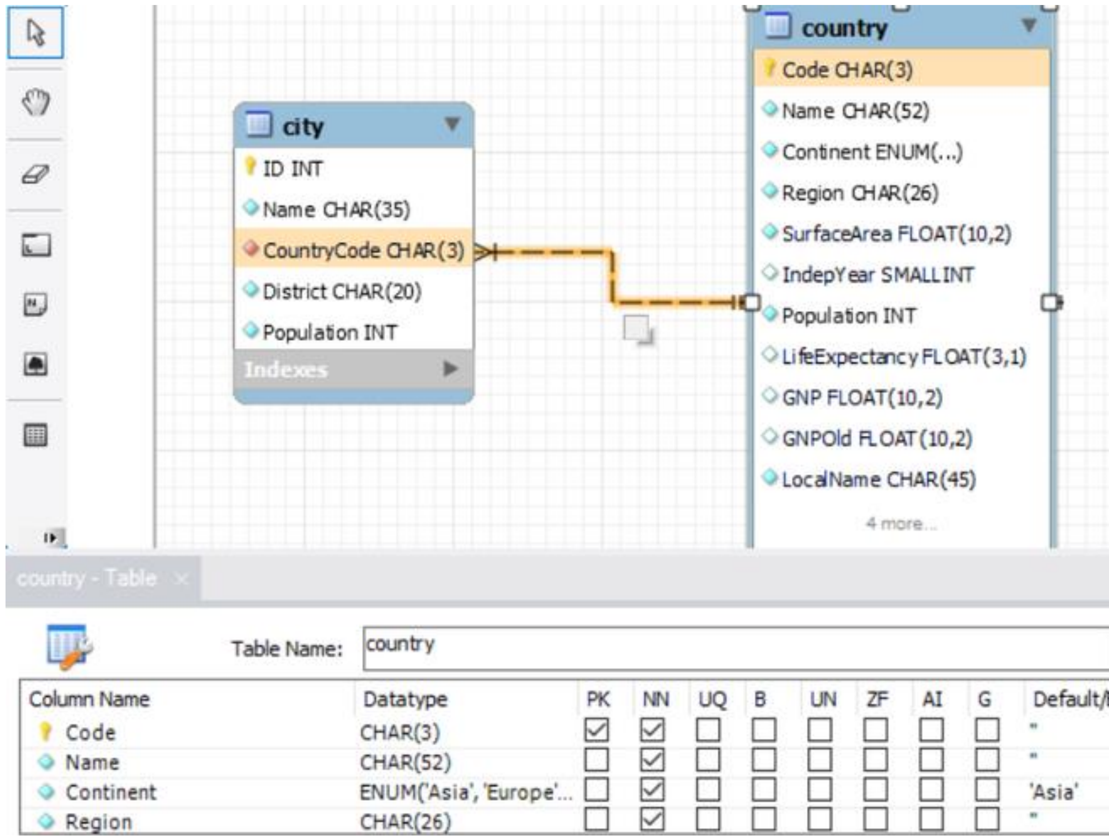
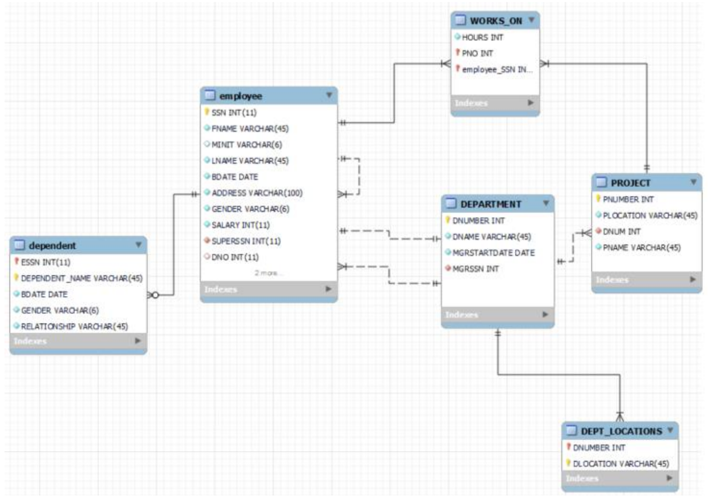


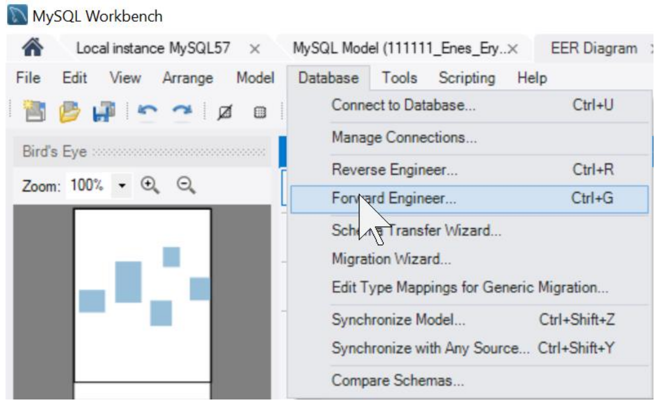
Table Name: country

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/
Code	CHAR(3)	✓	✓	☐	☐	☐	☐	☐	☐	''
Name	CHAR(52)	☐	✓	☐	☐	☐	☐	☐	☐	''
Continent	ENUM('Asia', 'Europe'...	☐	✓	☐	☐	☐	☐	☐	☐	'Asia'
Region	CHAR(26)	☐	✓	☐	☐	☐	☐	☐	☐	''

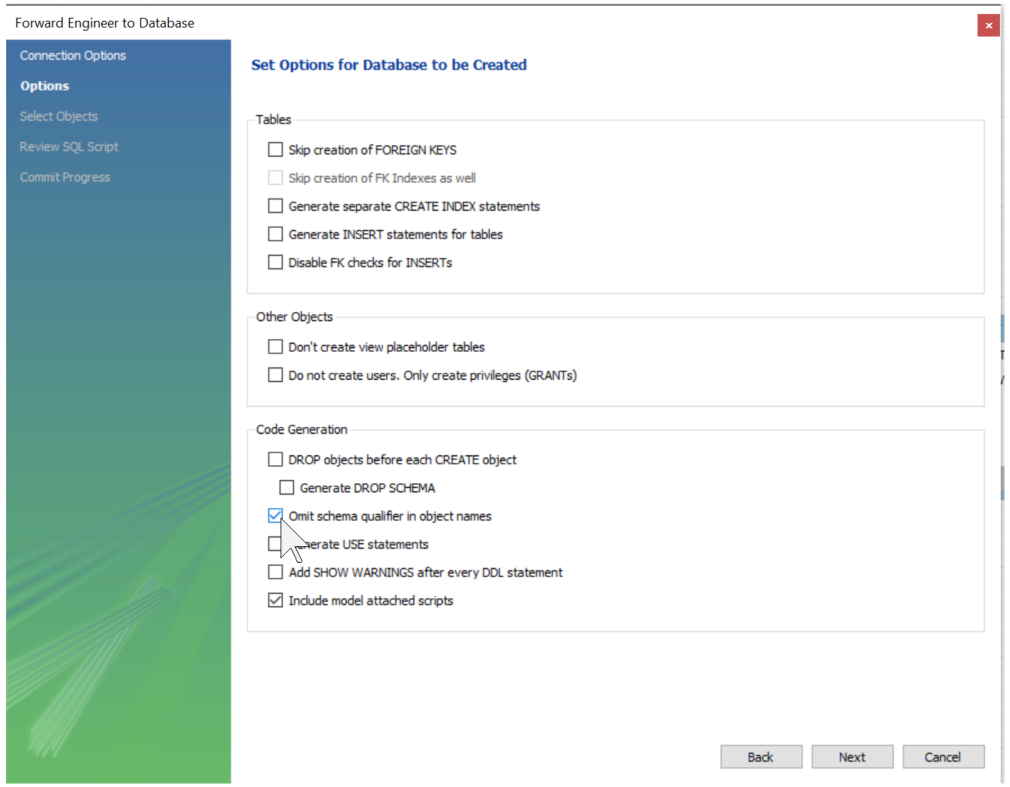
Creating database and entering data-2

- Step 2: Create the database using forward engineer:





MySQL Workbench
Local instance MySQL57
MySQL Model (111111_Enes_Ery...
EER Diagram
File
Edit
View
Arrange
Model
Database
Tools
Scripting
Help
Bird's Eye
Zoom: 100%
Connect to Database... Ctrl+U
Manage Connections...
Reverse Engineer... Ctrl+R
Forward Engineer... Ctrl+G
Schema Transfer Wizard...
Migration Wizard...
Edit Type Mappings for Generic Migration...
Synchronize Model... Ctrl+Shift+Z
Synchronize with Any Source... Ctrl+Shift+Y
Compare Schemas...


Forward Engineer to Database

Connection Options
Options
Select Objects
Review SQL Script
Commit Progress

Set Options for Database to be Created

Tables
- [ ] Skip creation of FOREIGN KEYS
- [ ] Skip creation of FK Indexes as well
- [ ] Generate separate CREATE INDEX statements
- [ ] Generate INSERT statements for tables
- [ ] Disable FK checks for INSERTs

Other Objects
- [ ] Don't create view placeholder tables
- [ ] Do not create users. Only create privileges (GRANTs)

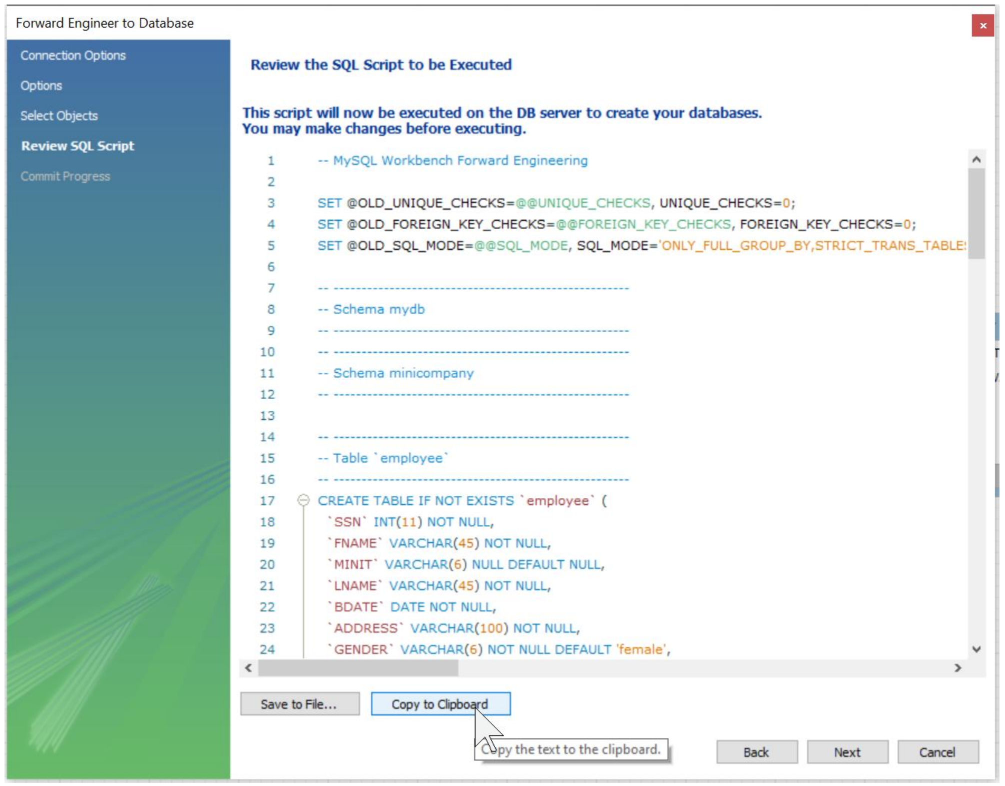
Code Generation
- [ ] DROP objects before each CREATE object
  - [ ] Generate DROP SCHEMA
- [✓] Omit schema qualifier in object names
- [ ] Generate USE statements
- [ ] Add SHOW WARNINGS after every DDL statement
- [✓] Include model attached scripts

Back Next Cancel

Make sure that you **select** «omit schema qualifier in object names»

Creating database and entering data-3

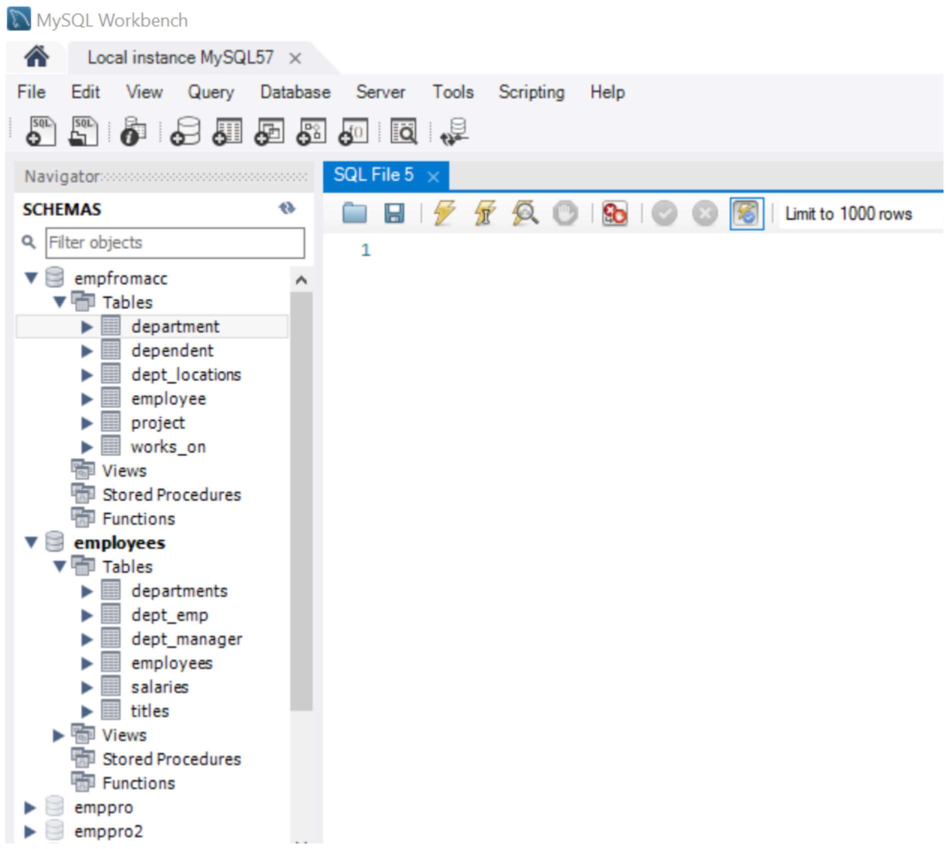
- Creating database and entering data-3
- Copy the code into clipboard, and hit CANCEL.
- Paste your code in a script window.



Review the SQL Script to be Executed

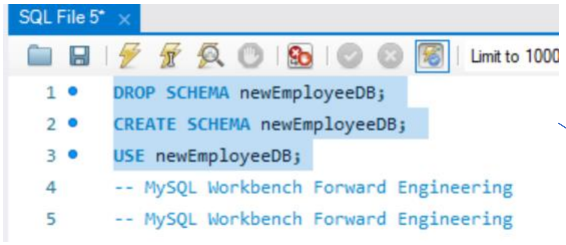
This script will now be executed on the DB server to create your databases.
You may make changes before executing.

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema minicompany
-- -----------------------------------------------------


-- -----------------------------------------------------
-- Table `employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `employee` (
  `SSN` INT(11) NOT NULL,
  `FNAME` VARCHAR(45) NOT NULL,
  `MINIT` VARCHAR(6) NULL DEFAULT NULL,
  `LNAME` VARCHAR(45) NOT NULL,
  `BDATE` DATE NOT NULL,
  `ADDRESS` VARCHAR(100) NOT NULL,
  `GENDER` VARCHAR(6) NOT NULL DEFAULT 'female',
```

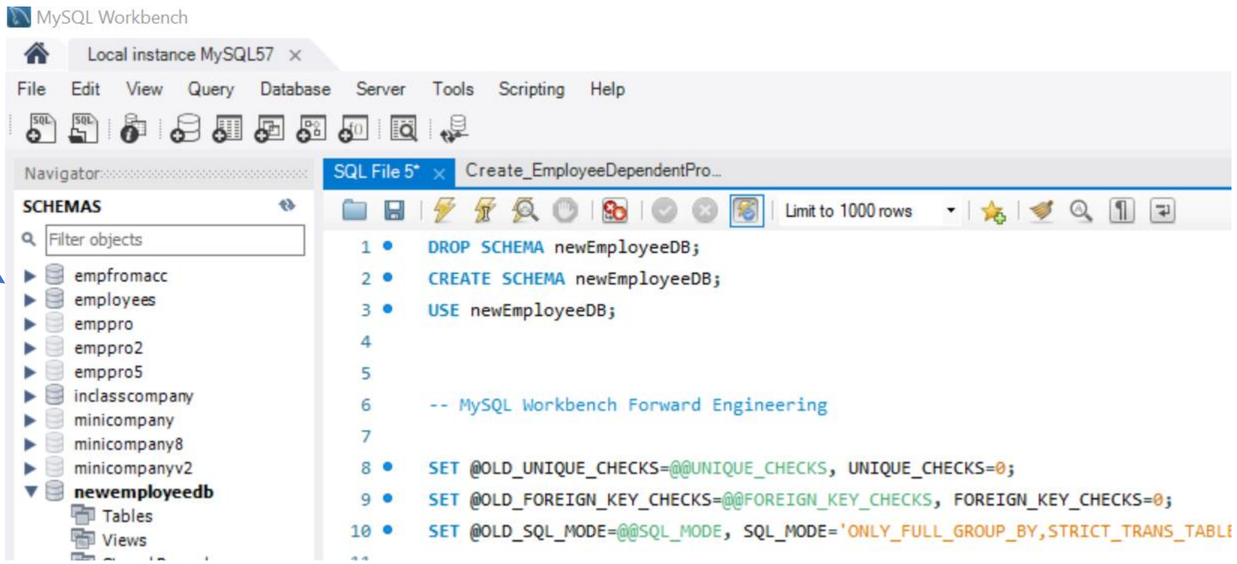


Creating database and entering data-4

- Add these three lines on the top, then run the query!



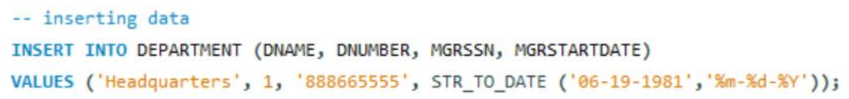
```
DROP SCHEMA newEmployeeDB;
CREATE SCHEMA newEmployeeDB;
USE newEmployeeDB;
-- MySQL Workbench Forward Engineering
-- MySQL Workbench Forward Engineering
```



```
DROP SCHEMA newEmployeeDB;
CREATE SCHEMA newEmployeeDB;
USE newEmployeeDB;


-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE
```

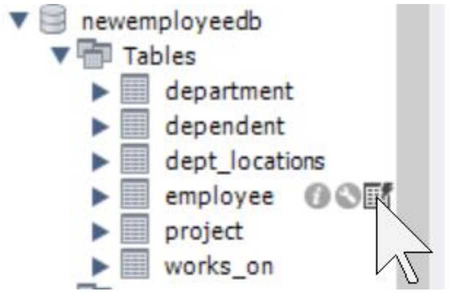
Creating database and entering data-4

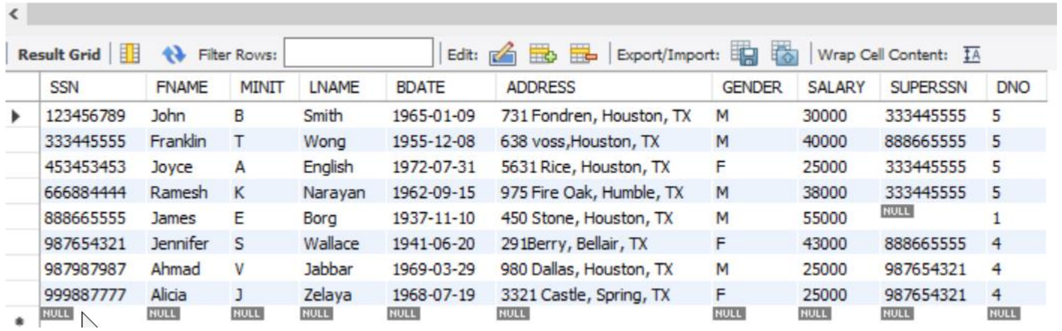
- Entering data:
  - Start with the EMPLOYEE data (figure out where to start by checking FK constraints)
- Option 1: using SQL STATEMENTS:



```
-- inserting data
INSERT INTO DEPARTMENT (DNAME, DNUMBER, MGRSSN, MGRSTARTDATE)
VALUES ('Headquarters', 1, '888665555', STR_TO_DATE ('06-19-1981','%m-%d-%Y'));
```

- Option 2: entering by hand using





SSN	FNAME	MINIT	LNAME	BDATE	ADDRESS	GENDER	SALARY	SUPERSSN	DNO
123456789	John	B	Smith	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
333445555	Franklin	T	Wong	1955-12-08	638 voss,Houston, TX	M	40000	888665555	5
453453453	Joyce	A	English	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
666884444	Ramesh	K	Narayan	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
888665555	James	E	Borg	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1
987654321	Jennifer	S	Wallace	1941-06-20	291Berry, Bellair, TX	F	43000	888665555	4
987987987	Ahmad	V	Jabbar	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
999887777	Alicia	J	Zelaya	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Options when running

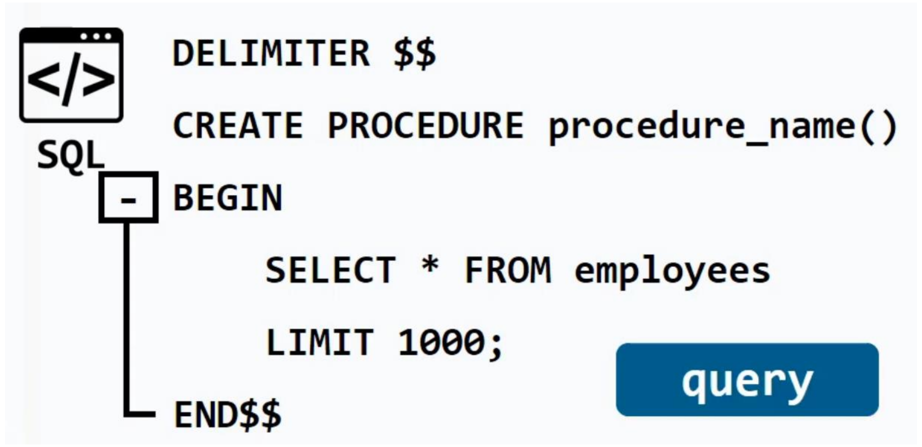
- **IN(…)** or **NOT IN(…)**
- **LIMIT(XYZ):** retireve only xyz many rows

Views

- A virtual table whose contents are obtained from existing table (aka. base table)
  - Protects the actual table
  - Not up-to-date
  - Faster (especially if many people use same portion of the base table in their queries)
- Must be **CREATE**d, and **REPLACE**d (updated)

Stored procedures/routines

- A usual action taking place periodically, or often.
- Aim: avoid writing the same chunk of code over an over again. Keep the code in database.
- Users can «call» the already written code (routine).
- Routine:
  - Function (user defined, besides the built-in ones)
  - Procedure ()
    - CREATE PROCEDURE procName()



```
DELIMITER $$
CREATE PROCEDURE procedure_name()
BEGIN
    SELECT * FROM employees
    LIMIT 1000;
END$$
```